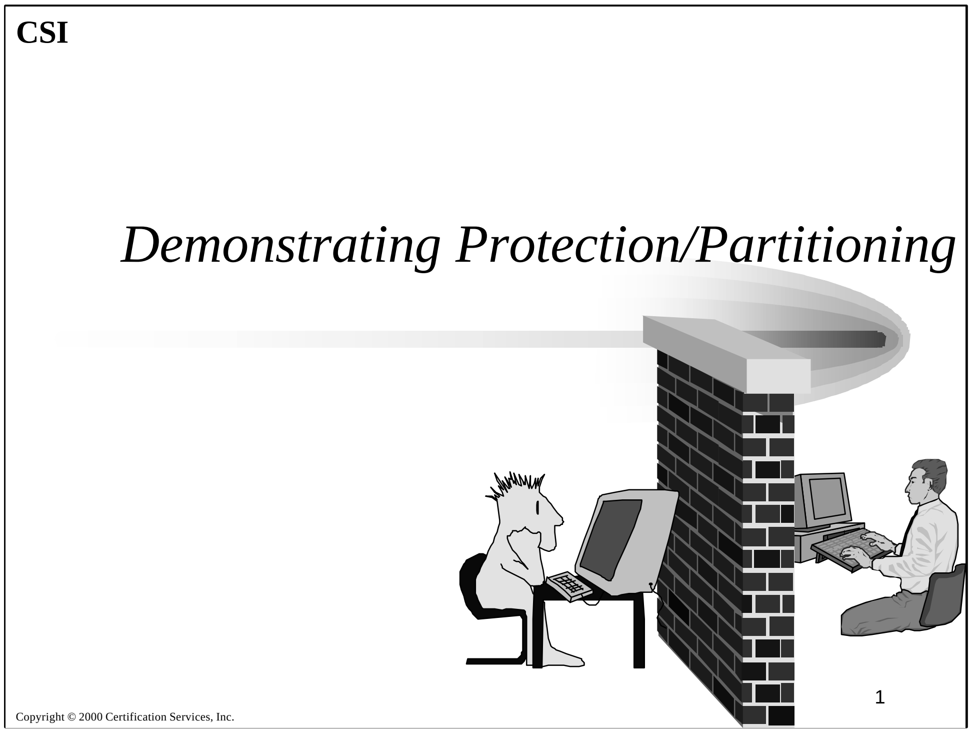
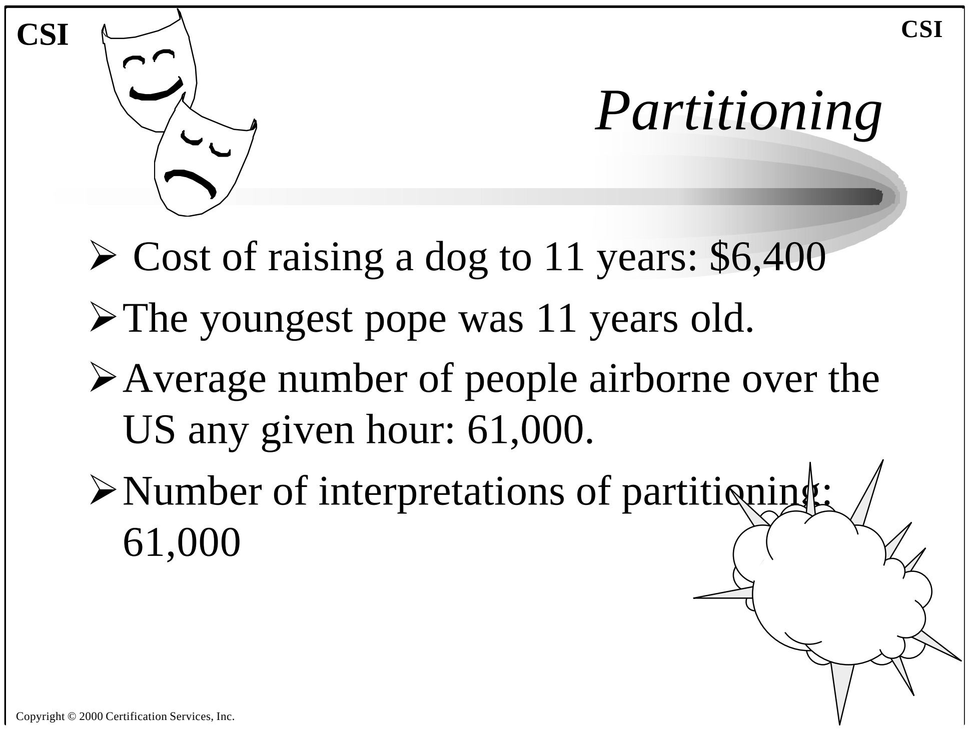


CSI

Demonstrating Protection/Partitioning

1

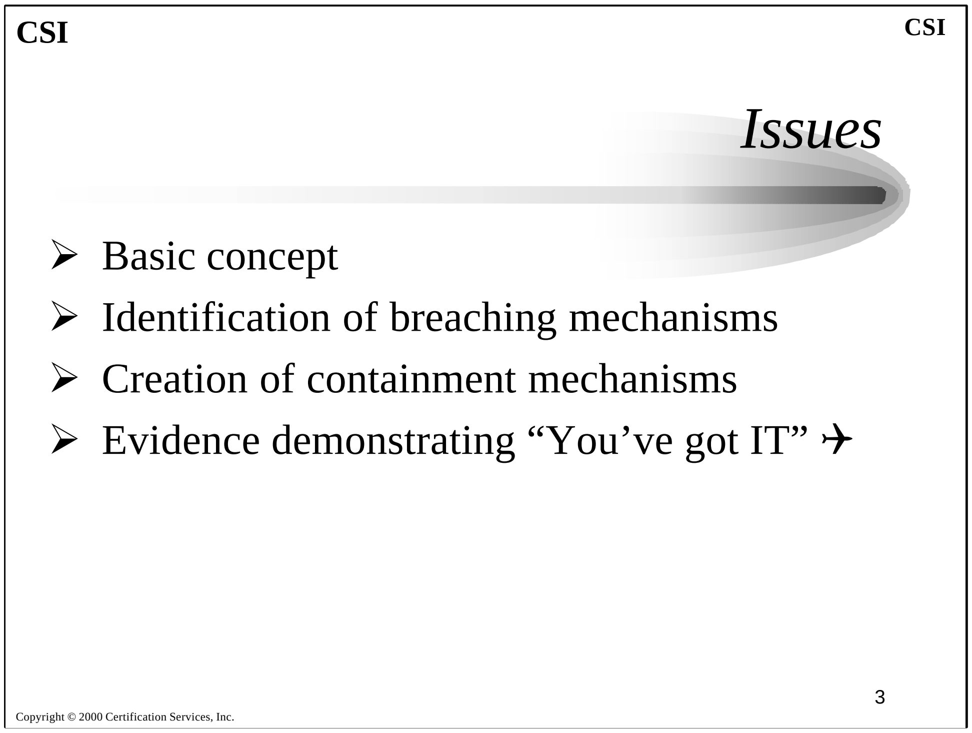


CSI

CSI

Partitioning

- Cost of raising a dog to 11 years: $6,400
- The youngest pope was 11 years old.
- Average number of people airborne over the US any given hour: 61,000.
- Number of interpretations of partitioning: 61,000



CSI CSI

Issues

- Basic concept
- Identification of breaching mechanisms
- Creation of containment mechanisms
- Evidence demonstrating “You’ve got IT” ✈

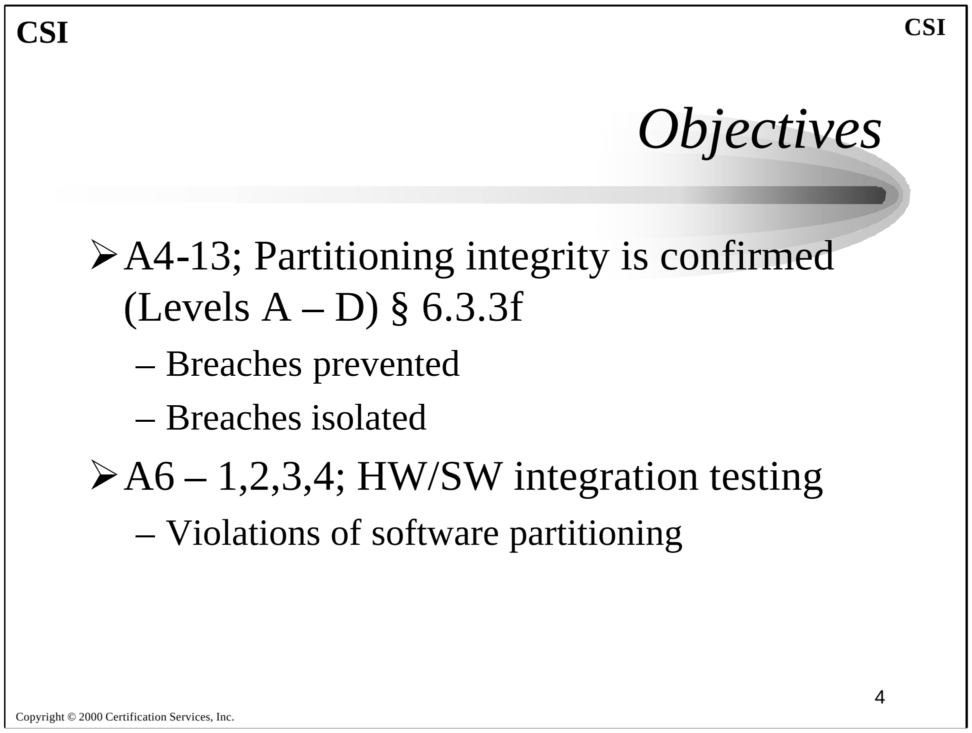
3

Copyright © 2000 Certification Services, Inc.

The basic concept of portioning is fairly simple – keep partitioned components from unacceptably interfering with each others behavior.

The first task is to identify all of the mechanisms that a components can interfere with the behavior of each other. In modern processors this is a very difficult task. Current approaches are very heuristic in nature and depend heavily on design engineers experience and expertise.

Once interference mechanisms have been identified then design features are required to eliminate or mitigate these Interferences.


CSI CSI

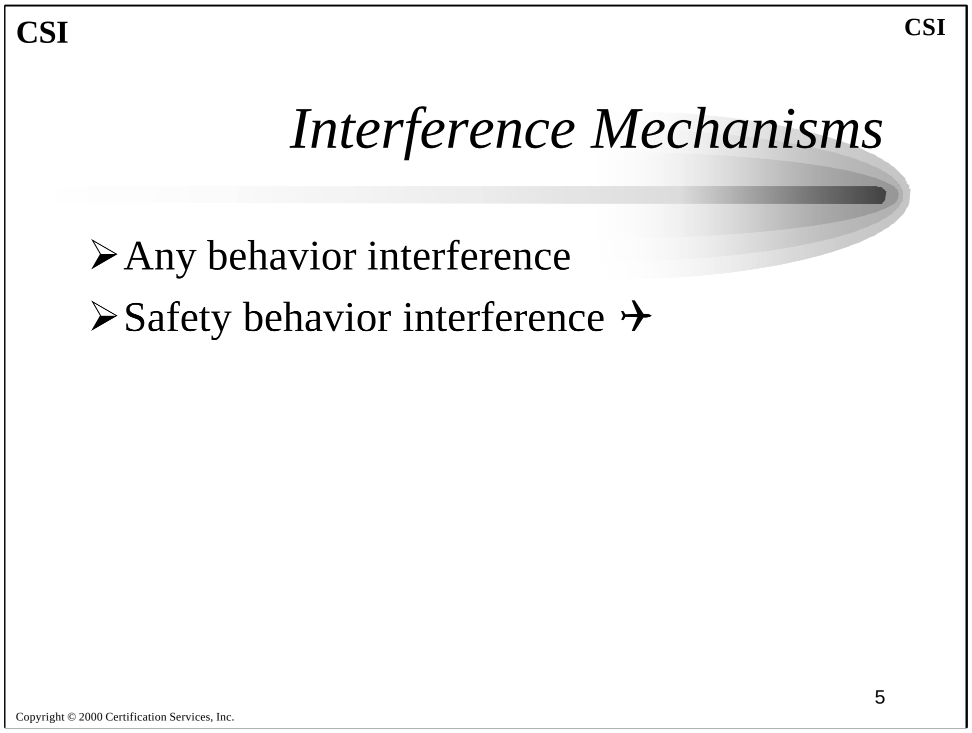
Objectives

- ➢ A4-13; Partitioning integrity is confirmed (Levels A – D) § 6.3.3f
  - – Breaches prevented
  - – Breaches isolated
- ➢ A6 – 1,2,3,4; HW/SW integration testing
  - – Violations of software partitioning

4

Copyright © 2000 Certification Services, Inc.

DO-178B has exactly one explicit objective dealing with verification of the partitioning design. In addition buried deep within the text (6.4.3a of the objectives in table A-6 (Testing) there is a reference to tests to check for violations of software partitioning. There are an additional 10 references to partitioning sprinkled throughout the document. Appendix A to this module contains a reproduction of all of the text that pertains to this concept. Unfortunately being able to prove partitioning has been implemented beyond very simple design architectures is very difficult especially in a single processor environment.



CSI CSI

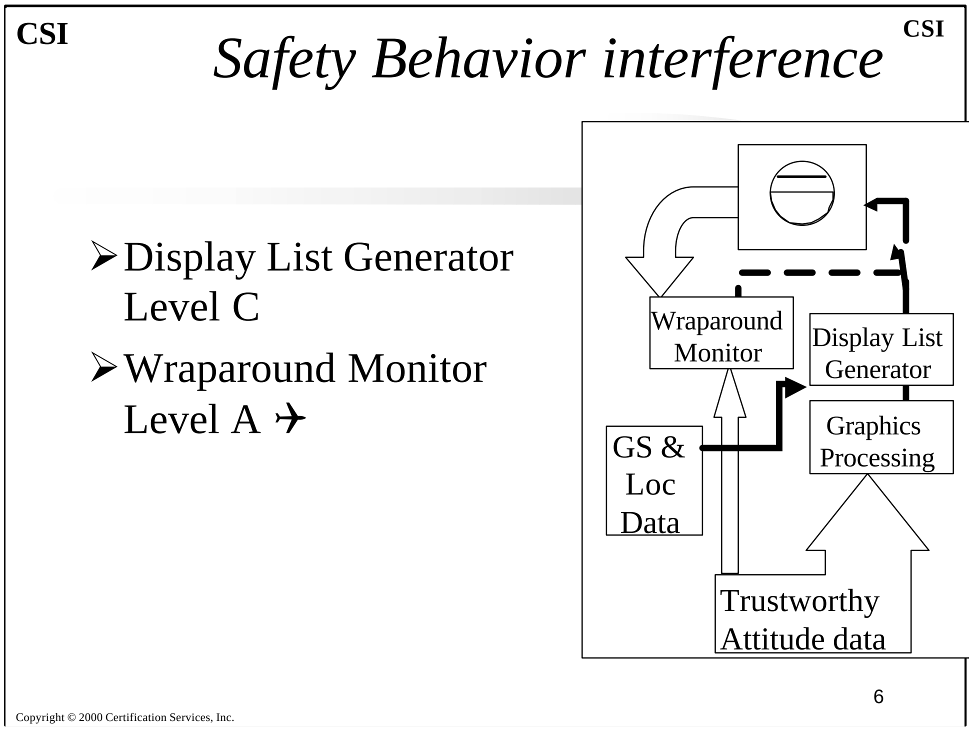
Interference Mechanisms

- ➢ Any behavior interference
- ➢ Safety behavior interference ✈

5

From a conceptual standpoint the easiest approach to establishing partitioning is to establish that no behavior of one component can have any effect on the behavior of another component. This is usually very expensive of resources. In the limit this leads to our standard non communicating federated avionics system (e.g. each system runs on its own processor and doesn not communicate with other systems. .

A more attractive approach is to relax our definition and only address interference mechanisms that interfere with the safety behavior of a component.

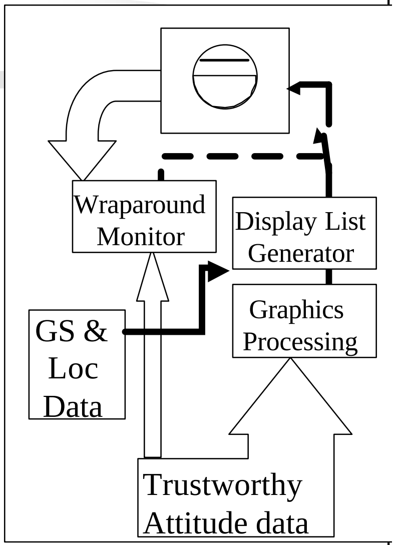


CSI

Safety Behavior interference

CSI

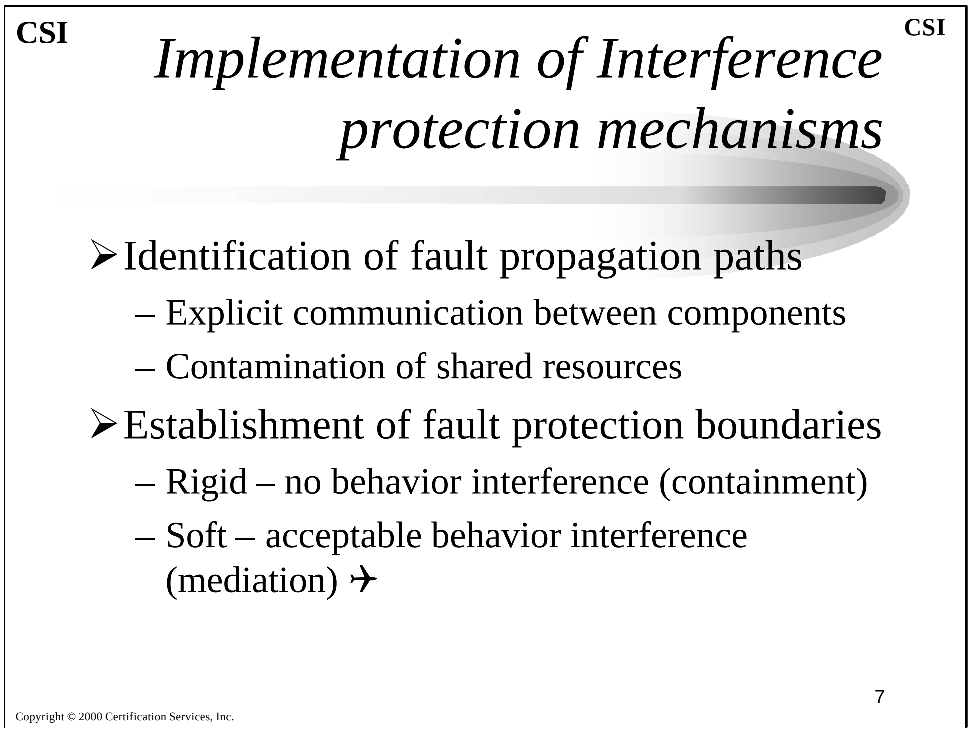
- ➢Display List Generator Level C
- ➢Wraparound Monitor Level A ✈



6

The attitude data is passed through a computer consisting of a display list generator and a graphics processor. A wrap around monitor is able read the LCD pixel settings and determine whether the attitude display matches the input attitude data. The localizer and GS data function creates display list data to display its information on the attitude display. If the Wraparound monitor is a hardware device then we can conclude that GS and Loc Data cannot interferewith the safety behavior of the display device.

If we had to guarantee that there was behavior interference we would need separate display hardware and communication channels for the attitude sections of the screen and the GS/Loc sections of the screen. The GS and LOC functions would probably have to be designed in a separate processor or have assigned memory and a given time slice ala round robin schedulers with no interrupt capability. .


CSI CSI

Implementation of Interference protection mechanisms

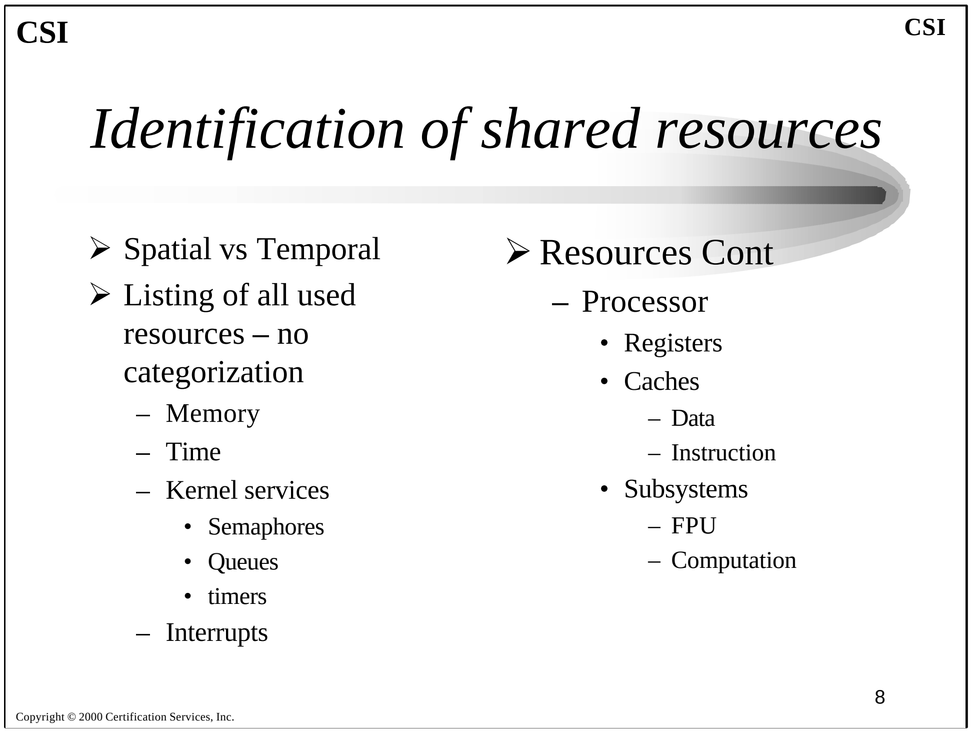
- ➢Identification of fault propagation paths
  - – Explicit communication between components
  - – Contamination of shared resources
- ➢Establishment of fault protection boundaries
  - – Rigid – no behavior interference (containment)
  - – Soft – acceptable behavior interference (mediation) ✈

7

Copyright © 2000 Certification Services, Inc.

The basic approach is to identify all of the ways that a fault could propagate from one component to another component. This can be divided up into to mechanisms. The first is where there is explicit communication between one component and another. This is usually the exchange of data. Even if two boxes were separated on different airplanes in different Line Replaceable units, the data communication via radio link could result in a fault in one component affecting the behavior of another component. The second is where two components share a common resource and rely on specific and correct behavior of the resource and one of the components contaminates that resource in a manner that the behavior of the other component is affected. An example of this would be memory that is shared between two components. A more subtle effect would be the cache system in modern high performance processors.

Once all of the propagation paths have been identified the fault protection boundaries can be established based on the desired amount of protection. If no behavior interference can be tolerated then rigid boundaries need to be devised (usually called fault containment boundaries). If instead the goal is to limit the behavior interference to acceptable outcomes then design strategies need to be developed to mediate any failures that arrive via a prorogation path.


CSI CSI

Identification of shared resources

- Spatial vs Temporal
- Listing of all used resources – no categorization
  - Memory
  - Time
  - Kernel services
    - Semaphores
    - Queues
    - timers
  - Interrupts
- Resources Cont
  - Processor
    - Registers
    - Caches
      - Data
      - Instruction
    - Subsystems
      - FPU
      - Computation

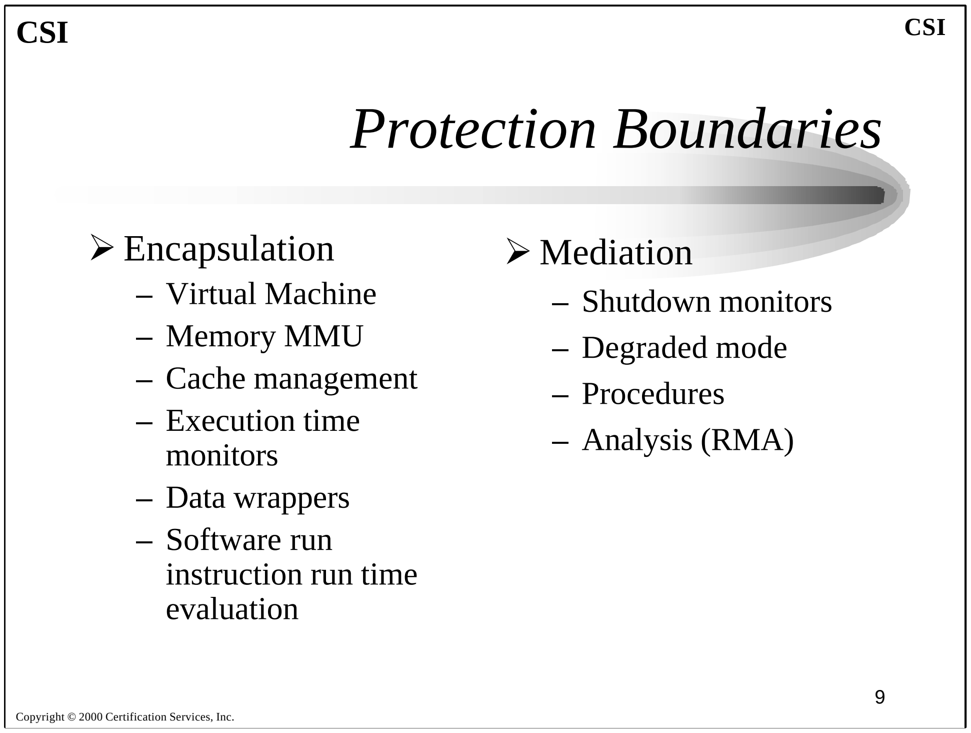
8

Copyright © 2000 Certification Services, Inc.

One approach that has been successfully used in the past is to categorize all shared resources as either spatial or temporal. The goal would be define a set of properties for each category such that if are satisfied would protection boundaries between components could be guaranteed. This has not been possible since this division does not appear to be orthogonal. For example data passed between two components that creates a timing error would be difficult to categorize. The normal approach is to provide a list of components within each category and then examine what should be done. This reduces an advantage for categorization.

The next approach is to list all of the resources a component depends upon. Then analyze all the potential methods that resource can be contaminated. This requires an intimate knowledge of how the computer works and well as any software abstraction layer.

For example If a system requires 5 ms of computation time every 50 msec then any mechanism that interfered with this could create a problem. If operating in the user mode certain undefined instructions could halt the machine from which a power up may be the only recovery. Since the component executes instructions on the computer a presumption is made that any bit patter could be presented to the instruction register and the behavior of the computer has to be determined. If this acceptable from a safety viewpoint then an architecural construct could be created to guarantee this type of behavior. The goal is to look at the design of the component and determine it's resources.

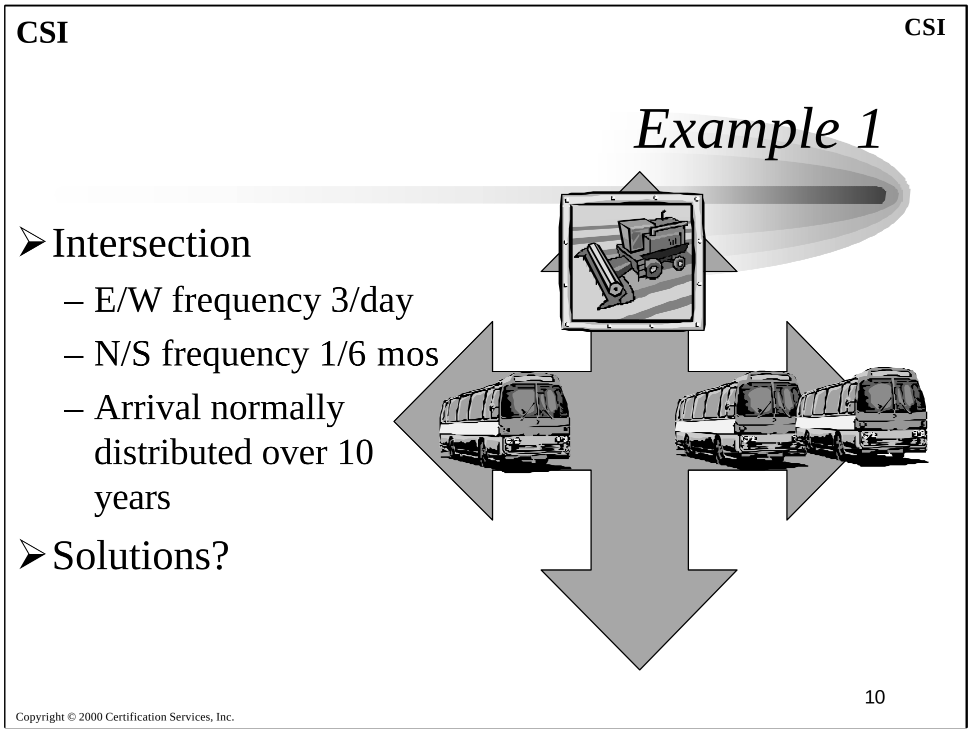


CSI CSI

Protection Boundaries

- Encapsulation
  - Virtual Machine
  - Memory MMU
  - Cache management
  - Execution time monitors
  - Data wrappers
  - Software run instruction run time evaluation
- Mediation
  - Shutdown monitors
  - Degraded mode
  - Procedures
  - Analysis (RMA)

CSI


CSI CSI

Example 1

- Intersection
  - E/W frequency 3/day
  - N/S frequency 1/6 mos
  - Arrival normally distributed over 10 years
- Solutions?

10

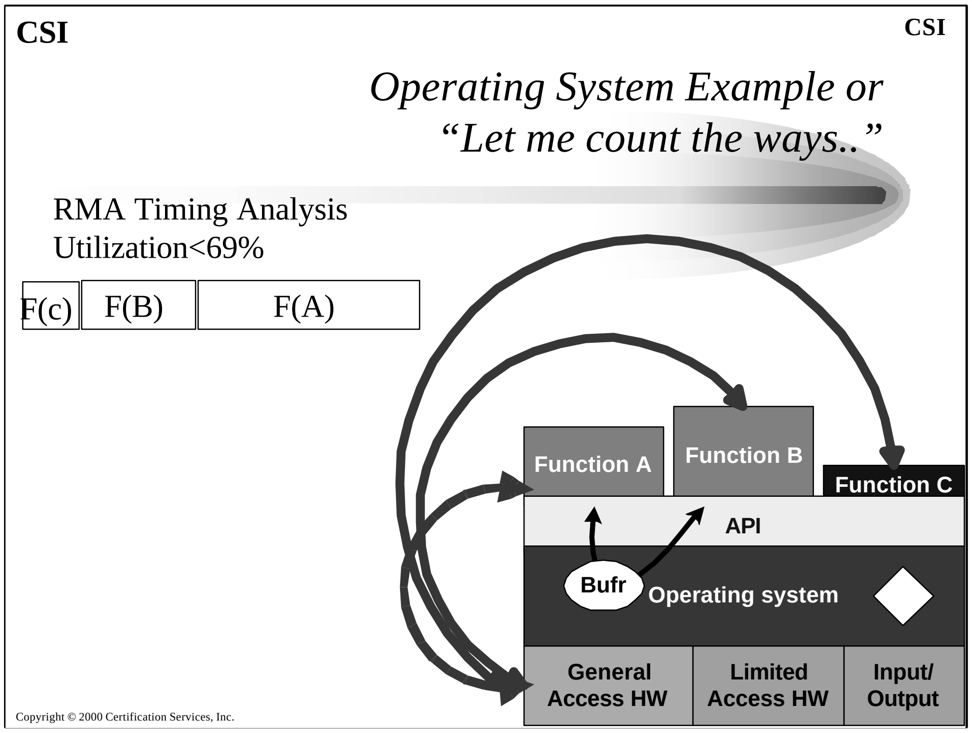
Copyright © 2000 Certification Services, Inc.

We have in intersection in the far flung farmlands of Iowa. As a result of a recent accident, authorities are trying to figure out how to handle the problem.

The East/West traffic needs the intersection to transit about 3 times/day. The North South traffic is typically some farm machinery with about 6 trailers in tow. I takes about 1 hour to transit the intersection assuming no failures.

For an encapsulation approach an overpass would be the answer.

For a mediation approach we would have to look at things like stop lights, local airports, a buffer zone for the stop light, or an acceptable statistical probability of an accident.

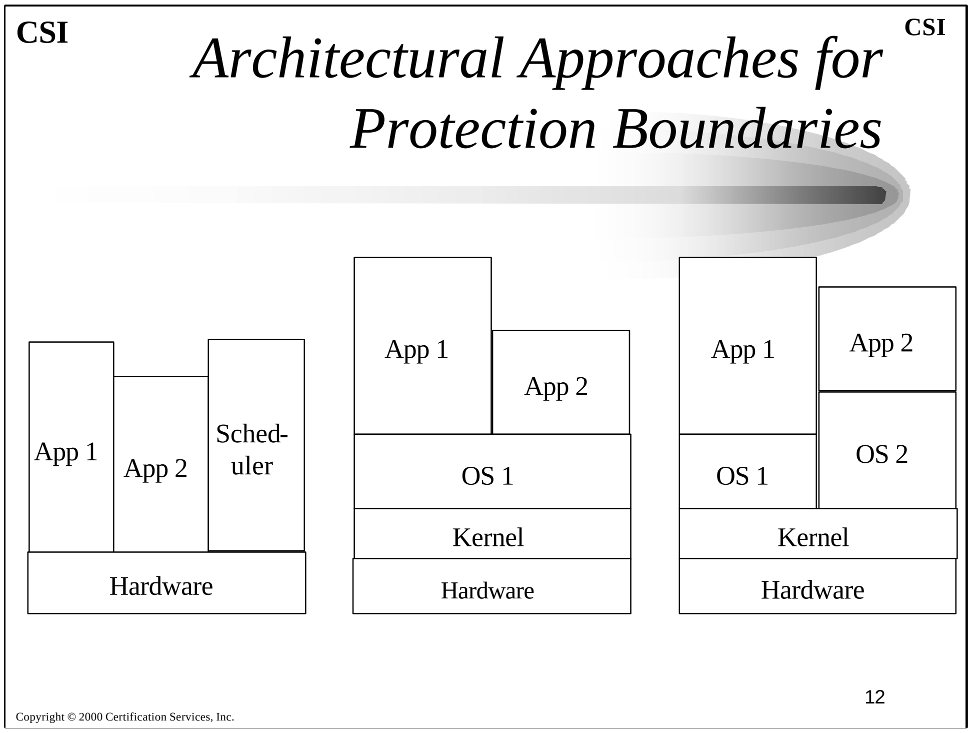

CSI CSI

Operating System Example or “Let me count the ways..”

RMA Timing Analysis
Utilization<69%

The system described here has three functions; Function A, Function B and Function C. The basic hardware has a memory management unit capability and a a Real Time Clock that occurs once every millisecond. The fastest application only executes at 20 HZ. The MMU ensures that each Function can only use the memory assigned to it’s partition. The only way to access the operating system or the input/output is through the API interface. The individual programs must execute on the computer but they only have access to a limited set of instructions by the design of the processor and the operating system. Function A and Function B can only communicate through a buffer controlled by the operating system. We used Rate Monotonic Analysis (RMA) to establish that all functions are schedulable based on the worst case timing analysis of each component.

There are a number of issues that could result in undesirable interference between the different component.s These will be jointly identified during the presentation jointly by the seminar participants and the presenter. The goal is to identify the fault propagation path for each function by examining both explicit and implicit communication mechanisms.



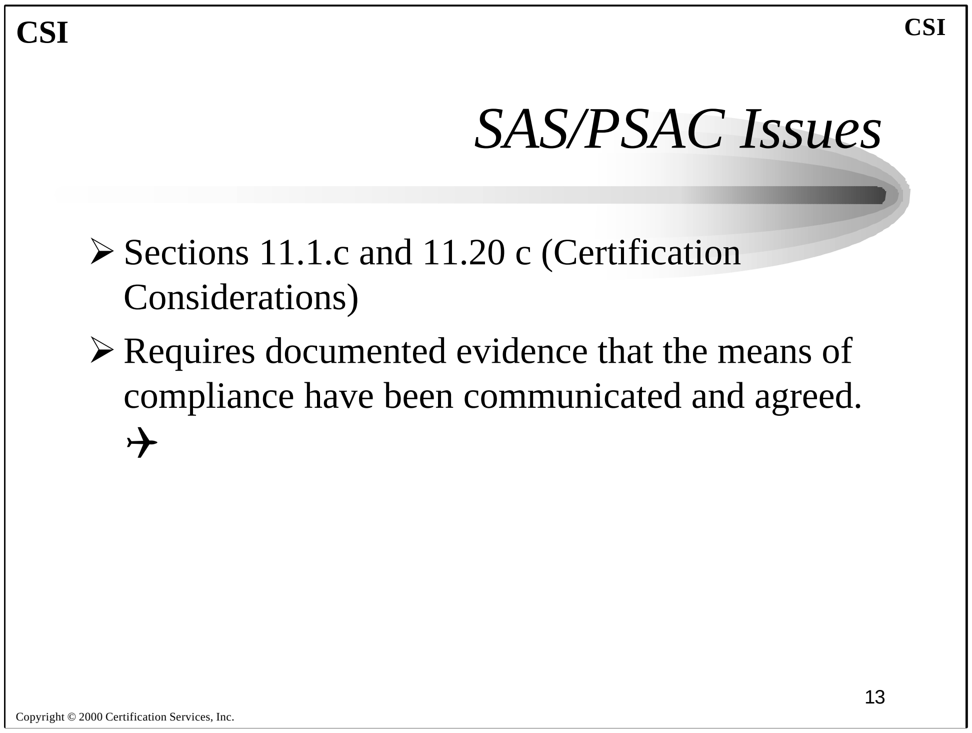
CSI CSI

Architectural Approaches for Protection Boundaries

12

Copyright © 2000 Certification Services, Inc.

In the first case each application has access to all of the hardware resources. In The second case a operating system with an API ensures that all calls from the Applications are handled by the OS before the hardware or any kernel resources can be affected. In the last example each application is given their own operating system. The operating systems can be customized for each application and could be different types as well (e.g. Windows 96 and LINUX). There is now a choice to develop the operating system to the highest level of assurance or to have the kernel provide the protections.

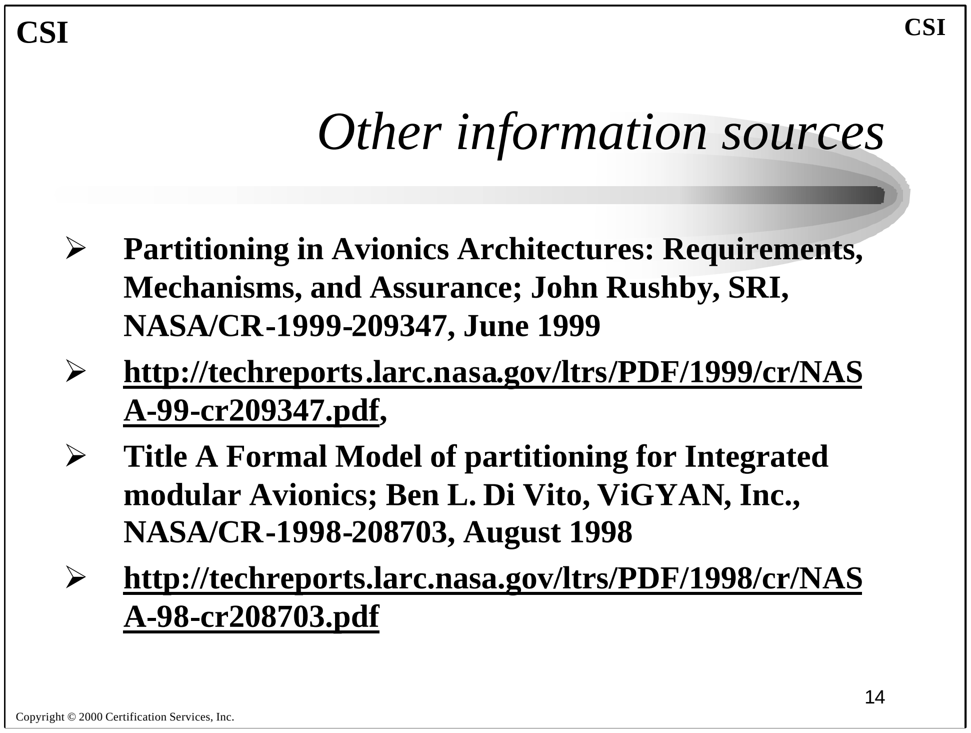


CSI CSI

SAS/PSAC Issues

- Sections 11.1.c and 11.20 c (Certification Considerations)
- Requires documented evidence that the means of compliance have been communicated and agreed. ✈

13

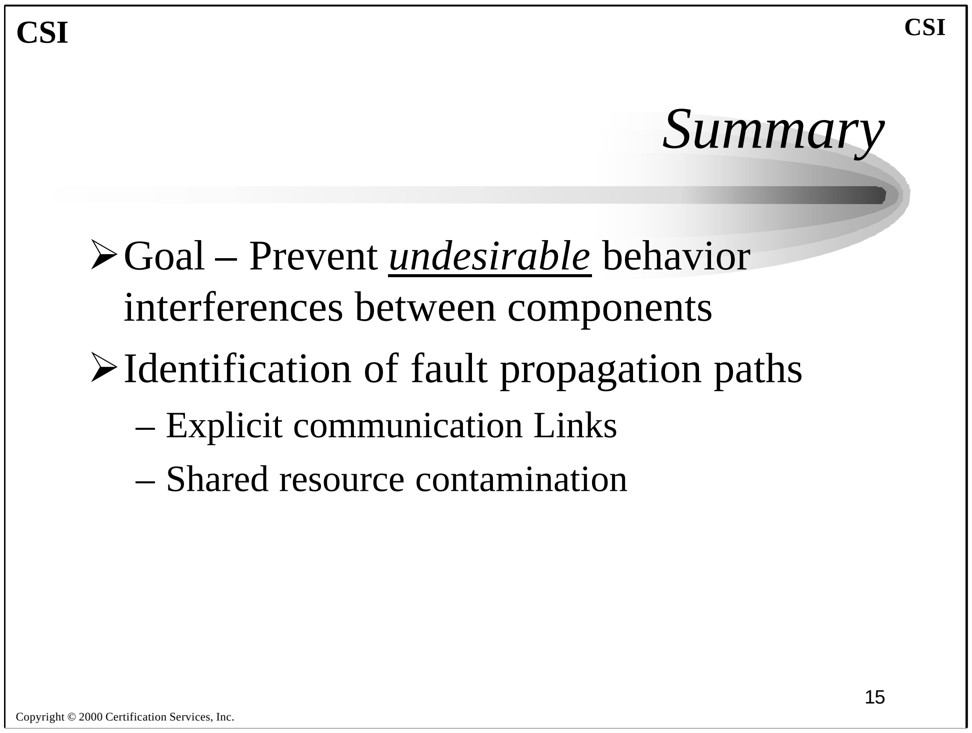


CSI CSI

Other information sources

- **Partitioning in Avionics Architectures: Requirements, Mechanisms, and Assurance; John Rushby, SRI, NASA/CR-1999-209347, June 1999**
- **http://techreports.larc.nasa.gov/ltrs/PDF/1999/cr/NASA-99-cr209347.pdf,**
- **Title A Formal Model of partitioning for Integrated modular Avionics; Ben L. Di Vito, ViGYAN, Inc., NASA/CR-1998-208703, August 1998**
- **http://techreports.larc.nasa.gov/ltrs/PDF/1998/cr/NASA-98-cr208703.pdf**

14



CSI CSI

Summary

- Goal – Prevent <u>*undesirable*</u> behavior interferences between components
- Identification of fault propagation paths
  - Explicit communication Links
  - Shared resource contamination

15